Ill-Posed Inverse Problems: Algorithms and Applications

Total Least Squares

Rosemary Renaut
with
Dr Hongbin Guo and Wolfgang Stefan

Department of Mathematics and Statistics
Arizona State University

February 16, 2006

Parameter Estimation Problem

- **Classical Approach** Linear Least Squares(A is exactly specified.)

$$x_{LS} = \arg\min_x ||Ax - b||_2^2$$

  - Orthogonal projection of b onto the range of A.

Parameter Estimation Problem

- **Classical Approach** Linear Least Squares(A is exactly specified.)

$$x_{LS} = \arg\min_{x} ||Ax - b||_2^2$$

  - Orthogonal projection of b onto the range of A.

- **Dense** A Form QR Decomposition of A, solve *directly* $Rx = Q^T b$.

Parameter Estimation Problem

- **Classical Approach** Linear Least Squares(A is exactly specified.)

$$x_{LS} = \arg\min_x ||Ax - b||_2^2$$

  - Orthogonal projection of b onto the range of A.
- **Dense** A Form QR Decomposition of A, solve *directly* $Rx = Q^T b$.
- **Sparse** A Use *iterative* techniques, CG, Krylov subspace, etc.

Integral Equations

$$\int_{\Omega} \text{input X system } d\Omega = \text{ output}$$

Integral Equations

$$\int_{\Omega} \text{input X system } d\Omega = \text{ output}$$

- Given noisy output determine input *or/and* the system.

Integral Equations

$$\int_{\Omega} \text{input X system } d\Omega = \text{ output}$$

- Given noisy output determine input *or/and* the system.
- **General Applications:** Image Processing, Signal Identification

Integral Equations

$$\int_\Omega \text{input X system } d\Omega = \text{ output}$$

- Given noisy output determine input *or/and* the system.
- **General Applications:** Image Processing, Signal Identification
- **Our Motivation** Seismic & Medical Signal Restoration

Invariant Model

- Signal degradation is modeled as a convolution

$$g = f \otimes h + n$$

  - where g is the blurred signal
  - f is the unknown signal
  - h is the point spread function (PSF)- **known**
  - n is noise

Invariant Model

- Signal degradation is modeled as a convolution

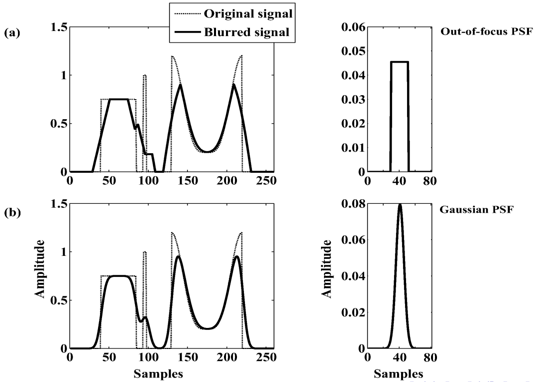
$$g = f \otimes h + n$$

  - where g is the blurred signal
  - f is the unknown signal
  - h is the point spread function (PSF)- **known**
  - n is noise
- **Matrix Formulation** H is Toeplitz (structured)

$$g = Hf + n$$

Example Of Convolution

$$g = f \otimes h + n$$



Inverse with Known PSF

- Find f from $\boxed{g = f \otimes h + n}$ given g and h with unknown n.

Inverse with Known PSF

- Find f from $\boxed{g = f \otimes h + n}$ given g and h with unknown n.
- Assuming normal distributed n yields the estimator

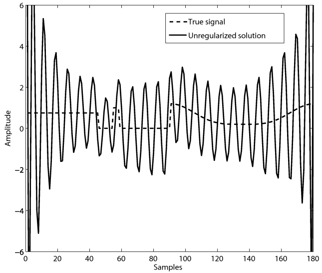
$$\hat{f} = \arg\min_f \{\|g - f \otimes h\|_2^2\}$$

Inverse with Known PSF

- Find f from $\boxed{g = f \otimes h + n}$ given g and h with unknown n.
- Assuming normal distributed n yields the estimator

$$\hat{f} = \arg\min_{f}\{\|g - f \otimes h\|_2^2\}$$

- Reconstruction with $n \in \mathbb{N}(0, 10^{-7})$



Ill-posedness

- Add more information about the signal

Ill-posedness

- Add more information about the signal
- e.g. statistical properties or information about the structure (e.g. sparse decon, or total variation decon)

Ill-posedness

- Add more information about the signal
- e.g. statistical properties or information about the structure (e.g. sparse decon, or total variation decon)
- Regularize

$$\hat{f} = \arg\min_{f}\{\|g - f \otimes h\|_2^2 + \lambda R(f)\},$$

where $R(f)$ is the penalty term and λ is a penalty parameter.

Standard Methods

- Tikhonov (TK).

$$R(f) = \mathrm{TK}(f) = \int_\Omega |\nabla f(x)|^2 dx.$$

Standard Methods

- Tikhonov (TK).

$$R(f) = \mathrm{TK}(f) = \int_{\Omega} |\nabla f(x)|^2 dx.$$

- Total Variation (TV)

$$R(f) = \mathrm{TV}(f) = \int_{\Omega} |\nabla f(x)| dx.$$

Standard Methods

- Tikhonov (TK).

$$R(f) = \mathrm{TK}(f) = \int_\Omega |\nabla f(x)|^2 dx.$$

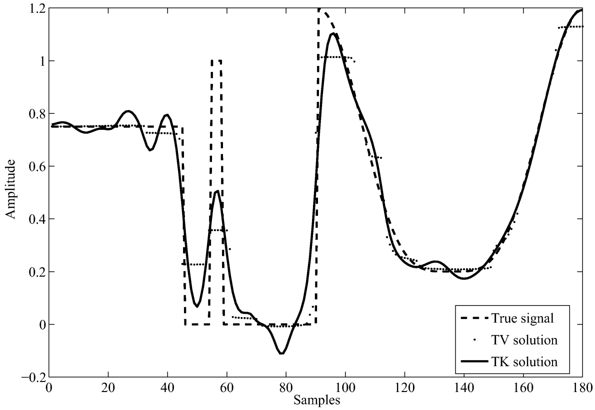
- Total Variation (TV)

$$R(f) = \mathrm{TV}(f) = \int_\Omega |\nabla f(x)| dx.$$

- Sparse deconvolution (L^1)

$$R(f) = \|f\|_1 = \int_\Omega |f(x)| dx.$$

Comparing TV and TK Regularization



Cost Functional

$$\hat{f} = \arg\min_{f}\{\|g - f \otimes h\|_2^2 + \lambda R(f)\}$$

- λ Governs the trade off between the fit to the data and the smoothness of the reconstruction and can be picked by the L-curve approach

Cost Functional

$$\hat{f} = \arg\min_{f}\{\|g - f \otimes h\|_2^2 + \lambda R(f)\}$$

- λ Governs the trade off between the fit to the data and the smoothness of the reconstruction and can be picked by the L-curve approach
- TV yields a piece wise constant reconstruction and preserves the edges of the signal

Cost Functional

$$\hat{f} = \arg\min_{f}\{\|g - f \otimes h\|_2^2 + \lambda R(f)\}$$

- λ Governs the trade off between the fit to the data and the smoothness of the reconstruction and can be picked by the L-curve approach
- TV yields a piece wise constant reconstruction and preserves the edges of the signal
- TK yields a smooth reconstruction

Cost Functional

$$\hat{f} = \arg\min_f \{ \|g - f \otimes h\|_2^2 + \lambda R(f) \}$$

- λ Governs the trade off between the fit to the data and the smoothness of the reconstruction and can be picked by the L-curve approach
- TV yields a piece wise constant reconstruction and preserves the edges of the signal
- TK yields a smooth reconstruction
- To find the minimum we use a limited memory BFGS method

Optimization

- TK is a linear least squares (LS) problem

$$\hat{f} = \arg\min_{f}\{\|g - Hf\|_2^2 + \lambda\|Lf\|_2^2)\}$$

Optimization

- TK is a linear least squares (LS) problem

$$\hat{f} = \arg\min_f \{\|g - Hf\|_2^2 + \lambda\|Lf\|_2^2)\}$$

- The TV objective function is **non differentiable**

$$J(f) = \|g - Hf\|_2^2 + \lambda\|Lf\|_1$$

Optimization

- TK is a linear least squares (LS) problem

$$\hat{f} = \arg\min_f \{\|g - Hf\|_2^2 + \lambda \|Lf\|_2^2)\}$$

- The TV objective function is **non differentiable**

$$J(f) = \|g - Hf\|_2^2 + \lambda \|Lf\|_1$$

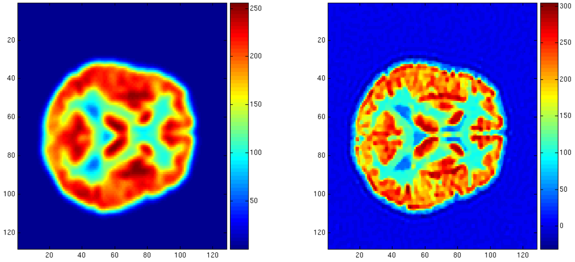
- **Evaluation** of the OF and its gradient **is cheap** (some FFTs and sparse matrix-vector multiplications)

Simulated PET

- Blur Segmented MRI scan using **Gaussian PSF**

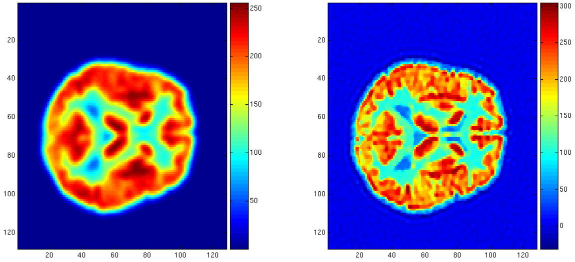
Simulated PET

- Blur Segmented MRI scan using **Gaussian PSF**
- Add Gauss noise to resulting Simulated PET image



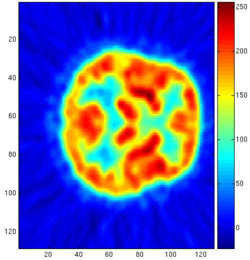
Simulated PET

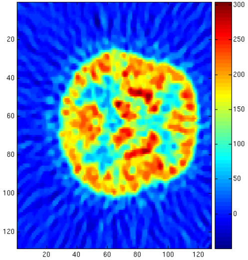
- Blur Segmented MRI scan using **Gaussian PSF**
- Add Gauss noise to resulting Simulated PET image



- **Note:** PSF is known regularize with TV

Real PET data





- Reconstruction done using Filtered Back Projection
- PSF estimated by a Gaussian
- TV regularization

Observations

- Image improvement is possible even with a rough estimation of the PSF (non-blind decon)

Observations

- Image improvement is possible even with a rough estimation of the PSF (non-blind decon)
- Total Variation regularization (piecewise constant solution) is appropriate: intensity levels depend on the tissue type.

Observations

- Image improvement is possible even with a rough estimation of the PSF (non-blind decon)
- Total Variation regularization (piecewise constant solution) is appropriate: intensity levels depend on the tissue type.
- Improvement requires better approximation of the PSF

Observations

- Image improvement is possible even with a rough estimation of the PSF (non-blind decon)
- Total Variation regularization (piecewise constant solution) is appropriate: intensity levels depend on the tissue type.
- Improvement requires better approximation of the PSF
- Increased Artifacts and noise. (More post processing can improve this)

Matrix Formulation: H not Toeplitz

- Consider $g = Hf + n$ where either H is structured but not known, or H is estimated and unstructured.

Matrix Formulation: H not Toeplitz

- Consider $g = Hf + n$ where either H is structured but not known, or H is estimated and unstructured.
- Example: Dynamic PET Estimate parameters $\mathcal{K} = (K_1, k_2, k_3, k_4)$, which satisfy

$$y(t) = u(t) \otimes \left(c_1(\mathcal{K}) e^{-\lambda_1(\mathcal{K})t} + c_2(\mathcal{K}) e^{-\lambda_2(\mathcal{K})t} \right).$$

Nonlinear parameter estimation can be converted to linear form

$$H = [\int(u); \int\int(u); \int(y); \int\int(y)],$$

$g = Hf$, but $f = f(\mathcal{K})$, is non linear.

- Statistics: Errors in variables regression(Gleser, 1981)

- Statistics: Errors in variables regression(Gleser, 1981)
  - Measurement error models (Fuller, 1987)
  - Orthogonal distance regression (Adcock, 1878)
  - Generalized least squares (Madansky, 1959,. . .)

- Statistics: Errors in variables regression(Gleser, 1981)
    - Measurement error models (Fuller, 1987)
    - Orthogonal distance regression (Adcock, 1878)
    - Generalized least squares (Madansky, 1959,. . .)
- Numerical Analysis: Golub & van Loan, 1981.

- Statistics: Errors in variables regression(Gleser, 1981)
  - Measurement error models (Fuller, 1987)
  - Orthogonal distance regression (Adcock, 1878)
  - Generalized least squares (Madansky, 1959,...)
- Numerical Analysis: Golub & van Loan, 1981.
- System Identification: Global linear least squares (Staar, 1981)

- Statistics: Errors in variables regression(Gleser, 1981)
  - Measurement error models (Fuller, 1987)
  - Orthogonal distance regression (Adcock, 1878)
  - Generalized least squares (Madansky, 1959,. . .)
- Numerical Analysis: Golub & van Loan, 1981.
- System Identification: Global linear least squares (Staar, 1981)
  - eigenvector method (Levin, 1964)
  - Koopmans-Levin method (Fernando & Nicholson, 1985)
  - Compensated least squares (Stoica & Soderstrom, 1982)

- Statistics: Errors in variables regression(Gleser, 1981)
  - Measurement error models (Fuller, 1987)
  - Orthogonal distance regression (Adcock, 1878)
  - Generalized least squares (Madansky, 1959,. . .)
- Numerical Analysis: Golub & van Loan, 1981.
- System Identification: Global linear least squares (Staar, 1981)
  - eigenvector method (Levin, 1964)
  - Koopmans-Levin method (Fernando & Nicholson, 1985)
  - Compensated least squares (Stoica & Soderstrom, 1982)
- Signal Processing: Minimum norm method (Kumaresan &Tufts, 1983)

$$\min \|(E, f)\|_F \quad \text{subject to} \quad (A+E)x = b+f$$

- Classical Algorithm Golub Reinsch Van-Loan

$$\text{Solve } [\hat{A}|\hat{b}] \begin{bmatrix} x_{TLS} \\ -1 \end{bmatrix} = 0, \ \hat{A} = A + E, \ \hat{b} = b + f.$$

$$\min \|(E, f)\|_F \quad \text{subject to} \quad (A+E)x = b+f$$

- Classical Algorithm Golub Reinsch Van-Loan

$$\text{Solve } [\hat{A}|\hat{b}] \begin{bmatrix} x_{TLS} \\ -1 \end{bmatrix} = 0, \ \hat{A} = A + E, \ \hat{b} = b + f.$$

- SVD Solution uses the SVD of augmented matrix $[A|b]$.

$$\begin{bmatrix} x_{TLS} \\ -1 \end{bmatrix} = \frac{-1}{v_{n+1,n+1}} v_{n+1}.$$

$\min \|(E, f)\|_F \quad \text{subject to} \quad (A+E)x = b+f$

- Classical Algorithm Golub Reinsch Van-Loan

$$\text{Solve } [\hat{A}|\hat{b}] \begin{bmatrix} x_{TLS} \\ -1 \end{bmatrix} = 0, \ \hat{A} = A + E, \ \hat{b} = b + f.$$

- SVD Solution uses the SVD of augmented matrix $[A|b]$.

$$\begin{bmatrix} x_{TLS} \\ -1 \end{bmatrix} = \frac{-1}{v_{n+1,n+1}} v_{n+1}.$$

- ▶ Direct Compute the SVD and solve.

Rayleigh Quotient Formulation for TLS

- An Iterative Approach:

$$x_{TLS} = \mathrm{argmin}_x \phi(x) = \mathrm{argmin}_x \frac{\|Ax - b\|^2}{1 + \|x\|^2},$$

Rayleigh Quotient Formulation for TLS

- An Iterative Approach:

$$x_{TLS} = \mathrm{argmin}_x \phi(x) = \mathrm{argmin}_x \frac{\|Ax - b\|^2}{1 + \|x\|^2},$$

- ϕ is the Rayleigh quotient of matrix $[A, b]$.

Rayleigh Quotient Formulation for TLS

- An Iterative Approach:

$$x_{TLS} = \mathrm{argmin}_x \phi(x) = \mathrm{argmin}_x \frac{\|Ax - b\|^2}{1 + \|x\|^2},$$

- ϕ is the Rayleigh quotient of matrix $[A, b]$.
- TLS minimizes the sum of squared normalized residuals.

An eigenvalue problem for TLS

- Björck, Hesternes and Matstoms(BHM), 2000 Eigenvalue equation for TLS

$$\begin{pmatrix} A^TA & A^Tb \\ b^TA & b^Tb \end{pmatrix} \begin{pmatrix} x \\ -1 \end{pmatrix} = \sigma^2 \begin{pmatrix} x \\ -1 \end{pmatrix}$$

An eigenvalue problem for TLS

- Björck, Hesternes and Matstoms(BHM), 2000 Eigenvalue equation for TLS
$$\begin{pmatrix} A^T A & A^T b \\ b^T A & b^T b \end{pmatrix} \begin{pmatrix} x \\ -1 \end{pmatrix} = \sigma^2 \begin{pmatrix} x \\ -1 \end{pmatrix}$$
- σ^2 is to be minimized. System matrix is constant.

An eigenvalue problem for TLS

- Björck, Hesternes and Matstoms(BHM), 2000 Eigenvalue equation for TLS

$$\begin{pmatrix} A^T A & A^T b \\ b^T A & b^T b \end{pmatrix} \begin{pmatrix} x \\ -1 \end{pmatrix} = \sigma^2 \begin{pmatrix} x \\ -1 \end{pmatrix}$$

- σ^2 is to be minimized. System matrix is constant.
- First block equation gives the normal equations for TLS

$$(A^T A - \sigma^2 I)x = A^T b.$$

An eigenvalue problem for TLS

- Björck, Hesternes and Matstoms(BHM), 2000 Eigenvalue equation for TLS

$$\begin{pmatrix} A^T A & A^T b \\ b^T A & b^T b \end{pmatrix} \begin{pmatrix} x \\ -1 \end{pmatrix} = \sigma^2 \begin{pmatrix} x \\ -1 \end{pmatrix}$$

- σ^2 is to be minimized. System matrix is constant.
- First block equation gives the normal equations for TLS

$$(A^T A - \sigma^2 I)x = A^T b.$$

- Use Rayleigh quotient iteration to find the minimal eigenvalue.

An eigenvalue problem for TLS

- Björck, Hesternes and Matstoms(BHM), 2000 Eigenvalue equation for TLS
$$\begin{pmatrix} A^T A & A^T b \\ b^T A & b^T b \end{pmatrix} \begin{pmatrix} x \\ -1 \end{pmatrix} = \sigma^2 \begin{pmatrix} x \\ -1 \end{pmatrix}$$
- σ^2 is to be minimized. System matrix is constant.
- First block equation gives the normal equations for TLS
$$(A^T A - \sigma^2 I)x = A^T b.$$
- Use Rayleigh quotient iteration to find the minimal eigenvalue.
- Solve *shifted normal equations* by Preconditioned Conjugate Gradients each iteration.

Tikhonov TLS

Regularization Stabilize TLS with realistic constraint on data $\|Lx\| \leq \delta$. δ is prescribed from known information on the solution, and L is typically a differential operator.

Tikhonov TLS

Regularization Stabilize TLS with realistic constraint on data $\|Lx\| \leq \delta$. δ is prescribed from known information on the solution, and L is typically a differential operator.

Reformulation with Lagrange Multiplier If the constraint is active the solution x^* satisfies *normal equations*

$$(A^TA + \lambda_I I + \lambda_L L^TL)x^* = A^Tb, \; (x^*)^T L^T L x^* - \delta^2 = 0,$$

$$\lambda_I = -\frac{\|Ax^* - b\|^2}{1 + \|x^*\|^2}, \quad \lambda_L = \mu(1 + \|x^*\|^2),$$

$$\mu = -\frac{1}{\delta^2}\left(\frac{b^T(Ax^* - b)}{1 + \|x^*\|^2} + \frac{\|Ax^* - b\|^2}{(1 + \|x^*\|^2)^2}\right),$$

Golub, Hansen and O'Leary, 1999. Solution technique is parameter dependent: λ_L, λ_I, δ, μ

Rayleigh Quotient Formulation for RTLS

- Recall

$$x_{TLS} = \mathrm{argmin}_x \phi(x) = \mathrm{argmin}_x \frac{\|Ax - b\|^2}{1 + \|x\|^2},$$

Rayleigh Quotient Formulation for RTLS

- Recall

$$x_{TLS} = \mathrm{argmin}_x \phi(x) = \mathrm{argmin}_x \frac{\|Ax - b\|^2}{1 + \|x\|^2},$$

- Formulate **regularization for TLS in RQ form**

$$\min_x \phi(x) \ \text{ subject to } \|Lx\| \leq \delta.$$

Rayleigh Quotient Formulation for RTLS

- Recall

$$x_{TLS} = \mathrm{argmin}_x \phi(x) = \mathrm{argmin}_x \frac{\|Ax - b\|^2}{1 + \|x\|^2},$$

- Formulate **regularization for TLS in RQ form**

$$\min_x \phi(x) \quad \text{subject to } \|Lx\| \leq \delta.$$

- Augmented Lagrangian

$$\mathcal{L}(x, \mu) = \phi(x) + \mu(\|Lx\|^2 - \delta^2).$$

Eigenvalue Problem RTLS: Renaut & Guo (SIAM 2004)

$$B(x^*)\begin{pmatrix} x^* \\ -1 \end{pmatrix} = -\lambda_I \begin{pmatrix} x^* \\ -1 \end{pmatrix},$$

$$B(x^*) = \begin{pmatrix} A^TA + \lambda_L(x^*)L^TL, & A^Tb \\ b^TA, & -\lambda_L(x^*)\gamma + b^Tb \end{pmatrix},$$

$$\lambda_L = \mu(1 + \|x^*\|^2),$$

Eigenvalue Problem RTLS: Renaut & Guo (SIAM 2004)

$$B(x^*)\begin{pmatrix} x^* \\ -1 \end{pmatrix} = -\lambda_I \begin{pmatrix} x^* \\ -1 \end{pmatrix},$$

$$B(x^*) = \begin{pmatrix} A^TA + \lambda_L(x^*)L^TL, & A^Tb \\ b^TA, & -\lambda_L(x^*)\gamma + b^Tb \end{pmatrix},$$

$$\lambda_L = \mu(1 + \|x^*\|^2),$$

- Seek the minimal eigenpair for B. Note B is not constant.

Eigenvalue Problem RTLS: Renaut & Guo (SIAM 2004)

$$B(x^*)\begin{pmatrix} x^* \\ -1 \end{pmatrix} = -\lambda_I \begin{pmatrix} x^* \\ -1 \end{pmatrix},$$

$$B(x^*) = \begin{pmatrix} A^TA + \lambda_L(x^*)L^TL, & A^Tb \\ b^TA, & -\lambda_L(x^*)\gamma + b^Tb \end{pmatrix},$$

$$\lambda_L = \mu(1 + \|x^*\|^2),$$

- Seek the minimal eigenpair for B. Note B is not constant.
- Specify constraint $\|Lx^*\|^2 \leq \delta^2$, use $\gamma = \delta^2$ or $\|Lx^*\|^2$

Development

- At a solution constraint is active: measure normalized discrepancy

$$g(x) = (\|Lx\|^2 - \delta^2)/(1 + \|x\|^2)$$

Development

- At a solution constraint is active: measure normalized discrepancy

$$g(x) = (\|Lx\|^2 - \delta^2)/(1 + \|x\|^2)$$

- Denote $\mathbf{B}(\theta) = M + \theta N$, $N = \begin{pmatrix} L^T L & 0 \\ 0 & -\delta^2 \end{pmatrix}$.

Development

- At a solution constraint is active: measure normalized discrepancy

$$g(x) = (\|Lx\|^2 - \delta^2)/(1 + \|x\|^2)$$

- Denote $\mathbf{B}(\theta) = M + \theta N$, $N = \begin{pmatrix} L^T L & 0 \\ 0 & -\delta^2 \end{pmatrix}$.
- Denote eigenpair for smallest eigenvalue of $\mathbf{B}(\theta)$ as $\varrho_{n+1}, (x_\theta^T, -1)^T$

Development

- At a solution constraint is active: measure normalized discrepancy

$$g(x) = (\|Lx\|^2 - \delta^2)/(1 + \|x\|^2)$$

- Denote $\mathbf{B}(\theta) = M + \theta N$, $N = \begin{pmatrix} L^T L & 0 \\ 0 & -\delta^2 \end{pmatrix}$.
- Denote eigenpair for smallest eigenvalue of $\mathbf{B}(\theta)$ as $\varrho_{n+1}, (x_\theta^T, -1)^T$
- Reformulation:

For a constant δ, find a θ such that $g(x_\theta) = 0$.

Matrix B

Lemma Assuming that $b^T A \neq 0$ and $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$, then the smallest eigenvalue of $\mathbf{B}(\theta)$ is simple.

Matrix B

Lemma Assuming that $b^T A \neq 0$ and $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$, then the smallest eigenvalue of $\mathbf{B}(\theta)$ is simple.

Lemma If $[A, b]$ is a full rank matrix, there exists one and only one positive number, denoted by θ^c, such that $\mathbf{B}(\theta^c)$ is singular, and

(i) the null eigenvalue of $\mathbf{B}(\theta^c)$ is simple
(ii) when $0 \leq \theta < \theta^c$, $\mathbf{B}(\theta)$ is positive definite, and
(iii) when $\theta > \theta^c$, $\mathbf{B}(\theta)$ has only one negative eigenvalue.

Uniqueness

Lemma If $b^T A \neq 0$, $[A, b]$ is full-rank, then solution of $g(x_\theta) = 0$ is unique and

(i) There exists a $\lambda_L^* \in [0, \theta^c]$ which solves $g(x_\theta) = 0$.
(ii) When $\lambda_L \in (0, \lambda_L^*)$, $g(x_{\lambda_L}) > 0$ and $\lambda_L \in (\lambda_L^*, \infty)$, $g(x_{\lambda_L}) < 0$.

Uniqueness

Lemma If $b^T A \neq 0$, $[A, b]$ is full-rank, then solution of $g(x_\theta) = 0$ is unique and

(i) There exists a $\lambda_L^* \in [0, \theta^c]$ which solves $g(x_\theta) = 0$.

(ii) When $\lambda_L \in (0, \lambda_L^*)$, $g(x_{\lambda_L}) > 0$ and $\lambda_L \in (\lambda_L^*, \infty)$, $g(x_{\lambda_L}) < 0$.

Observe We see from this result that there is an unique solution to our problem and that an algorithm for finding this solution should depend both on finding an update for the Lagrange parameter λ_L and monitoring the sign of $g(x_{\lambda_L})$.

The Update Equation for $\theta = \lambda_L$

$$\begin{aligned} \lambda_L^{(k+1)} &= \lambda_L^{(k)} + \iota^{(k)} \frac{\lambda_L^{(k)}}{\delta^2} g(x_{\lambda_L^{(k)}}), \\ 0 < \iota^{(k)} &\leq 1 \text{ such that } g(x_{\lambda_L^{(k+1)}}) g(x_{\lambda_L^{(0)}}) > 0. \end{aligned}$$

The Update Equation for $\theta = \lambda_L$

$$\begin{aligned}\lambda_L^{(k+1)} &= \lambda_L^{(k)} + \iota^{(k)} \frac{\lambda_L^{(k)}}{\delta^2} g(x_{\lambda_L^{(k)}}), \\ 0 < \iota^{(k)} &\leq 1 \text{ such that } g(x_{\lambda_L^{(k+1)}}) g(x_{\lambda_L^{(0)}}) > 0.\end{aligned}$$

Theorem **Given $\lambda_L^{(0)} > 0$ iteration converges to unique solution, λ_L^*.**

The Update Equation for $\theta = \lambda_L$

$$\lambda_L^{(k+1)} = \lambda_L^{(k)} + \iota^{(k)} \frac{\lambda_L^{(k)}}{\delta^2} g(x_{\lambda_L^{(k)}}),$$

$$0 < \iota^{(k)} \leq 1 \text{ such that } g(x_{\lambda_L^{(k+1)}}) g(x_{\lambda_L^{(0)}}) > 0.$$

Theorem **Given $\lambda_L^{(0)} > 0$ iteration converges to unique solution, λ_L^*.**

Observe $B(\lambda_L^{(k)})$ is always positive definite. For $0 < \lambda_L^{(0)} < \theta^c$ iteration is monotically increasing or decreasing dependent on $\lambda_L^{(0)} >< \lambda_L^*$.

EXACT RTLS: Alternating Iteration on λ_L and x.

Algorithm

Given δ, $\lambda_L^{(0)} > 0$, calculate eigenpair $(\varrho_{n+1}^{(0)}, x^{(0)})$. Set $k = 0$. Update $\lambda_L^{(k)}$ and $x^{(k)}$ until convergence.

1. While not converged **Do** $\iota^{(k)} = 1$

EXACT RTLS: Alternating Iteration on λ_L and x.

Algorithm

Given δ, $\lambda_L^{(0)} > 0$, calculate eigenpair $(\varrho_{n+1}^{(0)}, x^{(0)})$. Set $k = 0$. Update $\lambda_L^{(k)}$ and $x^{(k)}$ until convergence.

1. While not converged **Do** $\iota^{(k)} = 1$
   1.1 **Inner Iteration:** Until $g(x^{(k+1)})g(x^{(0)}) > 0$

EXACT RTLS: Alternating Iteration on λ_L and x.

Algorithm

Given δ, $\lambda_L^{(0)} > 0$, calculate eigenpair $(\varrho_{n+1}^{(0)}, x^{(0)})$. Set $k = 0$. Update $\lambda_L^{(k)}$ and $x^{(k)}$ until convergence.

1. While not converged **Do** $\iota^{(k)} = 1$
   1.1 **Inner Iteration:** Until $g(x^{(k+1)})g(x^{(0)}) > 0$
      1.1.1 Update $\lambda_L^{(k+1)}$.

EXACT RTLS: Alternating Iteration on λ_L and x.

Algorithm

Given δ, $\lambda_L^{(0)} > 0$, calculate eigenpair $(\varrho_{n+1}^{(0)}, x^{(0)})$. Set $k = 0$. Update $\lambda_L^{(k)}$ and $x^{(k)}$ until convergence.

1. While not converged **Do** $\iota^{(k)} = 1$
   - 1.1 **Inner Iteration:** Until $g(x^{(k+1)})g(x^{(0)}) > 0$
     - 1.1.1 Update $\lambda_L^{(k+1)}$.
     - 1.1.2 Calculate smallest eigenvalue, $\varrho_{n+1}^{(k)}$, and corresponding eigenvector, $[(x^{(k+1)})^T, -1]^T$, of matrix $\mathbf{B}(\lambda_L^{(k)})$.

EXACT RTLS: Alternating Iteration on λ_L and x.

Algorithm

Given δ, $\lambda_L^{(0)} > 0$, calculate eigenpair $(\varrho_{n+1}^{(0)}, x^{(0)})$. Set $k = 0$. Update $\lambda_L^{(k)}$ and $x^{(k)}$ until convergence.

1. While not converged **Do** $\iota^{(k)} = 1$
   - 1.1 **Inner Iteration:** Until $g(x^{(k+1)})g(x^{(0)}) > 0$
     - 1.1.1 Update $\lambda_L^{(k+1)}$.
     - 1.1.2 Calculate smallest eigenvalue, $\varrho_{n+1}^{(k)}$, and corresponding eigenvector, $[(x^{(k+1)})^T, -1]^T$, of matrix $\mathbf{B}(\lambda_L^{(k)})$.
     - 1.1.3 If $g(x^{(k+1)})g(x^{(0)}) < 0$, set $\iota^{(k)} = \iota^{(k)}/2$ else **Break**.

EXACT RTLS: Alternating Iteration on λ_L and x.

Algorithm

Given δ, $\lambda_L^{(0)} > 0$, calculate eigenpair $(\varrho_{n+1}^{(0)}, x^{(0)})$. Set $k = 0$. Update $\lambda_L^{(k)}$ and $x^{(k)}$ until convergence.

1. While not converged **Do** $\iota^{(k)} = 1$
   - 1.1 **Inner Iteration:** Until $g(x^{(k+1)})g(x^{(0)}) > 0$
     - 1.1.1 Update $\lambda_L^{(k+1)}$.
     - 1.1.2 Calculate smallest eigenvalue, $\varrho_{n+1}^{(k)}$, and corresponding eigenvector, $[(x^{(k+1)})^T, -1]^T$, of matrix $\mathbf{B}(\lambda_L^{(k)})$.
     - 1.1.3 If $g(x^{(k+1)})g(x^{(0)}) < 0$, set $\iota^{(k)} = \iota^{(k)}/2$ else **Break**.
   - 1.2 Test for convergence. If converged **Break** else $k = k + 1$.

EXACT RTLS: Alternating Iteration on λ_L and x.

Algorithm

Given δ, $\lambda_L^{(0)} > 0$, calculate eigenpair $(\varrho_{n+1}^{(0)}, x^{(0)})$. Set $k = 0$. Update $\lambda_L^{(k)}$ and $x^{(k)}$ until convergence.

1. While not converged **Do** $\iota^{(k)} = 1$
   - 1.1 **Inner Iteration:** Until $g(x^{(k+1)})g(x^{(0)}) > 0$
     - 1.1.1 Update $\lambda_L^{(k+1)}$.
     - 1.1.2 Calculate smallest eigenvalue, $\varrho_{n+1}^{(k)}$, and corresponding eigenvector, $[(x^{(k+1)})^T, -1]^T$, of matrix $\mathbf{B}(\lambda_L^{(k)})$.
     - 1.1.3 If $g(x^{(k+1)})g(x^{(0)}) < 0$, set $\iota^{(k)} = \iota^{(k)}/2$ else **Break**.
   - 1.2 Test for convergence. If converged **Break** else $k = k + 1$.
2. **End Do.** $x_{RTLS} = x^{(k)}$.

Inner Iteration Solve I

- Find eigenvector $y^{(k,j+1)} = ((x^{(k,j+1)})^T, -1)^T$ such that

$$\begin{aligned} \mathbf{B}(\lambda_L^{(k)}) y^{(k,j+1)} &= \beta_{(k,j)} y^{(k,j)}, \\ \mathbf{B}(\lambda_L^{(k)}) &= \begin{pmatrix} J^{(k,j)} & A^T b \\ b^T A & \eta_{(k,j)} \end{pmatrix}, \end{aligned}$$

$$J^{(k,j)} = A^T A + \lambda_L^{(k)} L^T L \qquad \eta_{(k,j)} = b^T b - \lambda_L^{(k)} \delta^2,$$

Inner Iteration Solve II

- Apply Block Gaussian Elimination:

$$\begin{pmatrix} J^{(j)} & A^T b \\ 0 & \tau_j \end{pmatrix} \begin{pmatrix} x^{(j+1)} \\ -1 \end{pmatrix} = \beta_j \begin{pmatrix} x^{(j)} \\ -(z^{(j)})^T x^{(j)} - 1 \end{pmatrix},$$

Inner Iteration Solve II

- Apply Block Gaussian Elimination:

$$\begin{pmatrix} J^{(j)} & A^T b \\ 0 & \tau_j \end{pmatrix} \begin{pmatrix} x^{(j+1)} \\ -1 \end{pmatrix} = \beta_j \begin{pmatrix} x^{(j)} \\ -(z^{(j)})^T x^{(j)} - 1 \end{pmatrix},$$

- where

$$\tau_j = \eta_j - b^T A z^{(j)} \quad x^{(j+1)} = z^{(j)} + \beta_j u^{(j)} \quad \beta_j = \tau_j / ((z^{(j)})^T x^{(j)} + 1).$$

Inner Iteration Solve II

- Apply Block Gaussian Elimination:

$$\begin{pmatrix} J^{(j)} & A^T b \\ 0 & \tau_j \end{pmatrix} \begin{pmatrix} x^{(j+1)} \\ -1 \end{pmatrix} = \beta_j \begin{pmatrix} x^{(j)} \\ -(z^{(j)})^T x^{(j)} - 1 \end{pmatrix},$$

- where

$$\tau_j = \eta_j - b^T A z^{(j)} \quad x^{(j+1)} = z^{(j)} + \beta_j u^{(j)} \quad \beta_j = \tau_j / ((z^{(j)})^T x^{(j)} + 1).$$

- $z^{(j)}$ and $u^{(j)}$ solve $J^{(j)} z^{(j)} = A^T b$, $J^{(j)} u^{(j)} = x^{(j)}$.

Practical Details

- **The Sign Condition:** If the condition $g(x^{(k+1)})g(x^{(0)}) > 0$ is relaxed a divergent sequence may result.

Practical Details

- **The Sign Condition:** If the condition $g(x^{(k+1)})g(x^{(0)}) > 0$ is relaxed a divergent sequence may result.
- **Inexact Update at Inner Iteration:** At the inner iteration it is crucial to maintain the sign condition on g. Hence it is sufficient to find approximate eigenpair for which the sign condition holds.

Practical Details

- **The Sign Condition:** If the condition $g(x^{(k+1)})g(x^{(0)}) > 0$ is relaxed a divergent sequence may result.
- **Inexact Update at Inner Iteration:** At the inner iteration it is crucial to maintain the sign condition on g. Hence it is sufficient to find approximate eigenpair for which the sign condition holds.
- **Do not shift** For inexact update we do not use the shift because the system matrix $\mathbf{B}$ is parameter dependent and it makes no practical sense to force convergence with the shift.

Practical Details

- **The Sign Condition:** If the condition $g(x^{(k+1)})g(x^{(0)}) > 0$ is relaxed a divergent sequence may result.
- **Inexact Update at Inner Iteration:** At the inner iteration it is crucial to maintain the sign condition on g. Hence it is sufficient to find approximate eigenpair for which the sign condition holds.
- **Do not shift** For inexact update we do not use the shift because the system matrix $\mathbf{B}$ is parameter dependent and it makes no practical sense to force convergence with the shift.
- **Choice of γ:** The theory is developed for $\gamma = \delta^2$ but the algorithm can use $\gamma = \|Lx^{k,j}\|^2$. If blow up does not occur, the algorithm converges more quickly.

Computational Considerations: Generalized SVD

- All approaches need to solve normal equations $Jw = f$,

$$(A^T A + \lambda_L L^T L)w = f$$

Computational Considerations: Generalized SVD

- All approaches need to solve normal equations $Jw = f$,

$$(A^TA + \lambda_L L^TL)w = f$$

- GSVD For the augmented matrix $[A, L]$, $2m^2n + 15n^3$ flops.

$$A = U \begin{pmatrix} \Sigma & 0 \\ 0 & I_{n-p} \end{pmatrix} X^{-1}, \;\; L = V \begin{pmatrix} M & 0 \end{pmatrix} X^{-1}.$$

U, V, $m \times n$ and $p \times p$, resp., orthonormal. X, $n \times n$ nonsingular.Σ, M diag., $p \times p$, entries σ_i μ_i, resp. Then

$$\left[\begin{pmatrix} \Sigma^2 & 0 \\ 0 & I_{n-p} \end{pmatrix} + \lambda_L \begin{pmatrix} M^2 & 0 \\ 0 & 0 \end{pmatrix} \right] X^{-1}w = X^Tf.$$

Computational Considerations: Generalized SVD

- All approaches need to solve normal equations $Jw = f$,

$$(A^T A + \lambda_L L^T L)w = f$$

- GSVD For the augmented matrix $[A, L]$, $2m^2 n + 15n^3$ flops.

$$A = U \begin{pmatrix} \Sigma & 0 \\ 0 & I_{n-p} \end{pmatrix} X^{-1}, \;\; L = V \begin{pmatrix} M & 0 \end{pmatrix} X^{-1}.$$

  U, V, $m \times n$ and $p \times p$, resp., orthonormal. X, $n \times n$ nonsingular.Σ, M diag., $p \times p$, entries σ_i μ_i, resp. Then

$$\left[\begin{pmatrix} \Sigma^2 & 0 \\ 0 & I_{n-p} \end{pmatrix} + \lambda_L \begin{pmatrix} M^2 & 0 \\ 0 & 0 \end{pmatrix} \right] X^{-1} w = X^T f.$$

- Efficient Direct solution can be found efficiently $8n^2$ flops. Solve triangular systems for each λ_L.

Inverse Problems

Total Least Squares Methods

Results

Experiments

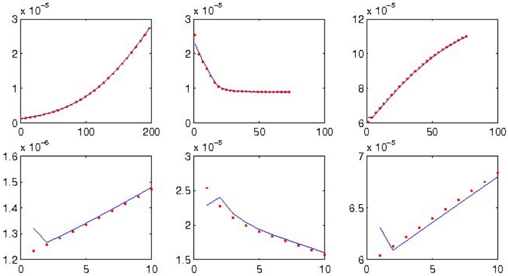
Conclusions

Other Research

Experiments: Hansen's Regularization Toolbox

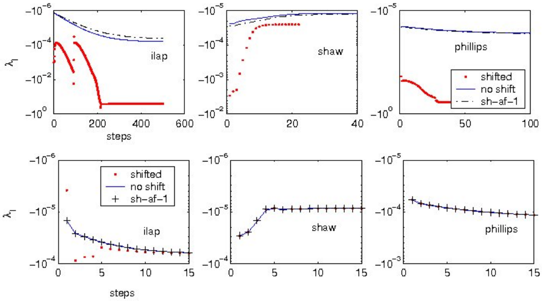
- *ilaplace, phillips* and *shaw* -all discretizations of continuous ill-posed Fredholm integral type constructed by quadrature.
- Generate A and x^*, calculate $b = Ax^*$ exactly.
- Scale $\|A\|_F = \|b\|_2 = 1$.
- Add noise 5% Gaussian to b and A.
- L approxs first derivative, $(n-1) \times n$.
- Tolerance $\tau = 10^{-4}$.
- Estimated solution x_{est}. Relative error with respect to x^*.
- Number of system solves K.

Inexact and Exact Algorithms: convergence for $-\lambda_l^{(k)}$



The dotted and dashed lines show convergence for exact and inexact algorithms, resp.. The first row shows the whole convergence history while the second row shows the first 10 steps. Left to right *ilaplace*, *shaw* and *phillips* resp..

Inclusion of Shift: convergence history of $-\lambda_I$

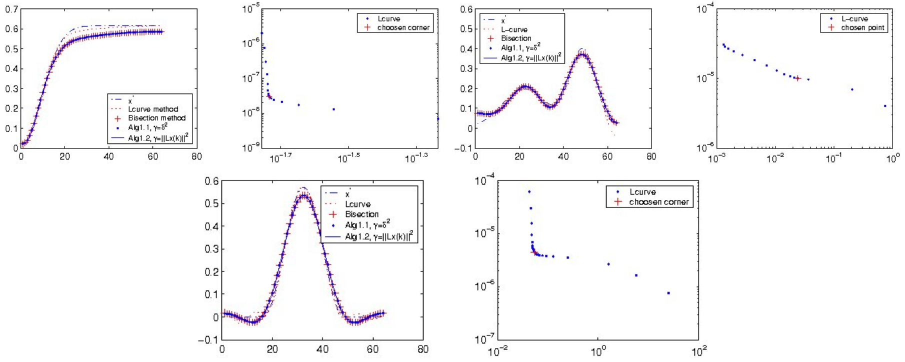


Using $\gamma = \delta^2$ and $\gamma = \|Lx^{k,j}\|^2$. top and bottom, resp.

shifted, no shift, or shift after first step.

ilaplace, *shaw* and *phillips* resp..

Comparison for γ: δ^2 or $\|Lx^{(k,j)}\|^2$



On top *ilaplace*, and *phillips*. Below *shaw*. Solutions are indicated on the left, and the L-curve on the right.

Inverse Problems

Total Least Squares Methods

Results

Conclusions

Other Research

Regularized Total Least Squares

Numerical experiments have been presented which verify the theory

- The algorithm provides an efficient and practical approach for the solution of the RTLS problem in which a good estimate of the physical parameter is provided.

Regularized Total Least Squares

Numerical experiments have been presented which verify the theory

- The algorithm provides an efficient and practical approach for the solution of the RTLS problem in which a good estimate of the physical parameter is provided.
- If blow up occurs bisection search may yield a better solution satisfying the constraint condition.

Regularized Total Least Squares

Numerical experiments have been presented which verify the theory

- The algorithm provides an efficient and practical approach for the solution of the RTLS problem in which a good estimate of the physical parameter is provided.
- If blow up occurs bisection search may yield a better solution satisfying the constraint condition.
- If no constraint information is provided, the L-curve technique can be successfully employed.

Regularized Total Least Squares

Numerical experiments have been presented which verify the theory

- The algorithm provides an efficient and practical approach for the solution of the RTLS problem in which a good estimate of the physical parameter is provided.
- If blow up occurs bisection search may yield a better solution satisfying the constraint condition.
- If no constraint information is provided, the L-curve technique can be successfully employed.
- Algorithm performs better than QEP for all of our tests.

Related Work

- Domain decomposition (PVDTLS) for large scale TLS: (2005).

Related Work

- Domain decomposition (PVDTLS) for large scale TLS: (2005).
- Applications :
  - Signal and image restoration (Geophysical Journal 2006)

Related Work

- Domain decomposition (PVDTLS) for large scale TLS: (2005).
- Applications :
  - Signal and image restoration (Geophysical Journal 2006)
  - Microarray analysis-pattern recognition: Support Vector Machine

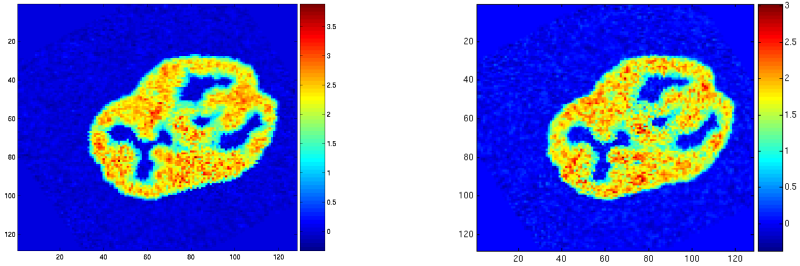
Related Work

- Domain decomposition (PVDTLS) for large scale TLS: (2005).
- Applications :
  - Signal and image restoration (Geophysical Journal 2006)
  - Microarray analysis-pattern recognition: Support Vector Machine
- Current Directions
  - Domain Decomposition for large scale RTLS

Related Work

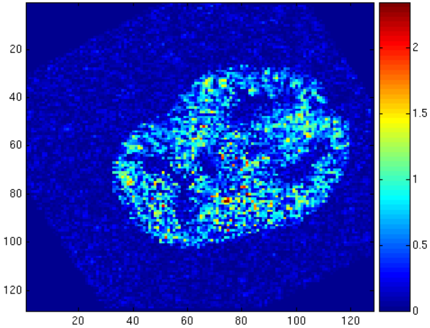
- Domain decomposition (PVDTLS) for large scale TLS: (2005).
- Applications :
  - Signal and image restoration (Geophysical Journal 2006)
  - Microarray analysis-pattern recognition: Support Vector Machine
- Current Directions
  - Domain Decomposition for large scale RTLS
  - Total Variation TLS
  - Large scale implementation
  - Multiple Right Hand Sides.

Removal of Noise from Difference Images for Longitudinal Studies



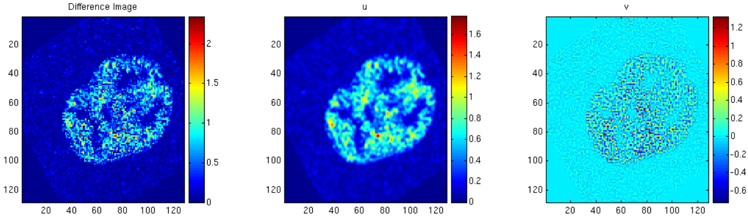
- **Application:** Two PET scans of the same patient at different times
- **Question:** Are there any anatomical or functional changes?

Difference Image after Alignment by Maximization of Mutual Information

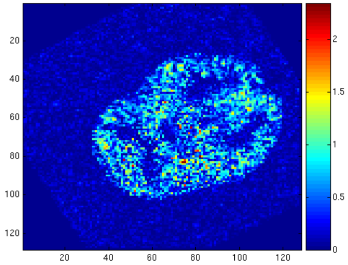


- Scans from different days have to be aligned
- Noise and artifacts change from scan to scan.
- Small changes are hard to locate in the difference image

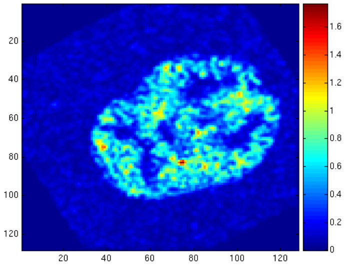
Decomposed Difference Image: wavelets



Difference Image and u Part

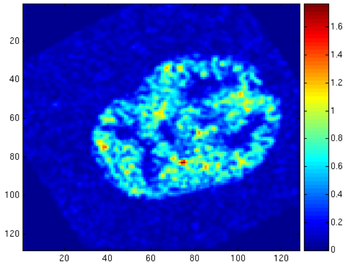


Difference Image

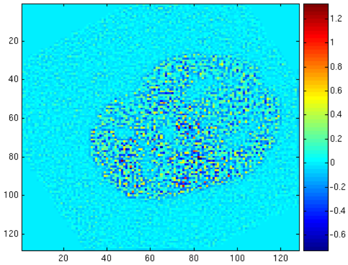


u Part

u and v Part



u Part



v Part

THANKYOU

Any Questions